

1. Objetivo General

Desarrollar una aplicación que permita manejar la descripción del caso expuesto en el punto 4.

2. Objetivos Específicos

- 2.1. Aplicar los conceptos del modelo conceptual y relacional.
- 2.2. Crear una Base de Datos en MS Server.
- 2.3. Crear un Webservice/REST Service.
- 2.4. Crear una página Web.
- 2.5. Usar herramientas como AngularJs, Bootstrap, HTML5, CSS, Entity Framework, y reporting services o cristal reports.
- 2.6. Crear un plan de instalación.
- 2.7. Crear un plan de proyecto.

3. Datos Generales

- 3.1. El valor del proyecto: 25%
- 3.2. **Nombre código: TECAirlines**
- 3.3. La tarea debe ser implementada grupos de 4 personas.
- 3.4. La **fecha de entrega:**
 - 3.4.1. Plan de trabajo: 2/Abril/2019
 - 3.4.2. Resumen Ejecutivo Avance 1: 9/Abril/2019
 - 3.4.3. Resumen Ejecutivo Avance 2: 16/Abril/2019
 - 3.4.4. Resumen Ejecutivo Avance 3: 23/Abril/2019
 - 3.4.5. Funcionalidad completa: 27/ Abril /2019.
 - 3.4.6. Todos los 'subject' del envío debe ser: CE3101_P1_PT o CE3101_P1_RE[1-4] y deben enviarse al correo mriveramTec@gmail.com.
- 3.5. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.

4. Descripción del caso de estudio

La empresa TEC-Airlines es una empresa en el mercado de transporte aéreo de personas de bajo costo y surgió luego de una larga negociación de la mejor universidad pública de Costa Rica con las aerolíneas comerciales del área, su objetivo principal es proveer a los estudiantes una alternativa más para que logren cumplir en especial sus becas en el exterior.

Por tanto TEC-Airlines se ve en la obligación de proveer a sus clientes las mismas facilidades que presenta una aerolínea comercial (aunque su mercado meta son los estudiantes no se restringe a ellos únicamente).

Reservaciones, Ésta es la plataforma que permitirá a los clientes realizar todas las funciones normales de una página web de una aerolínea, buscar vuelos, reservar vuelos, pagar sus vuelos, realizar el pre-chequeo

Aeropuerto, Ésta es la plataforma que permitirá a los funcionarios de la aerolínea poder realizar la apertura y cierre de vuelos, confirmación de reservaciones, asignación de maletas a un usuario etc.

4.1. Requerimientos del Software

4.1.1. Reservaciones.

- 4.1.1.1. *Registrar Usuario*: el sistema debe permitir la creación de distintos usuarios del sistema. Esta funcionalidad será utilizada por cualquier cliente de la aerolínea donde se solicitará información de contacto (Nombre completo, teléfono, correo y si es estudiante en caso de serlo solicitara la información de la Universidad donde estudia y su carnet que se utilizara para almacenar millas en su programa de lealtad)
- 4.1.1.2. *Búsqueda de vuelos*: los pasajeros tendrán la oportunidad de buscar sus vuelos ingresando su aeropuerto de origen y su aeropuerto de llegada.
- 4.1.1.3. *Reservación de vuelos*: Una vez seleccionado el vuelo se procederá con la reserva y se le habilitará la opción de cancelar con tarjeta de crédito.

4.1.2. Aeropuerto.

- 4.1.2.1. *Registrar Usuario*: Véase el punto 4.1.1.1.
- 4.1.2.2. *Búsqueda de vuelos*: Véase el punto 4.1.1.2.
- 4.1.2.3. *Reservación de vuelos*: Véase el punto 4.1.1.3.
- 4.1.2.4. *Promociones*: existirá un área dentro del sitio exclusivo para anunciar las promociones de la aerolínea.
- 4.1.2.5. *Chequeo de pasajeros*: esta funcionalidad permitirá a los pasajeros/funcionarios del aeropuerto chequear un pasajero en un vuelo, debe ser flexible y poder seleccionar el lugar de preferencia en el avión (asiento) y finalmente imprimir el pase de abordar que puede: ser enviado por correo, enviado a una impresora o un dispositivo móvil.
- 4.1.2.6. *Registro de vuelos*: para poder registrar un vuelo se debe seleccionar la ruta completa desde el aeropuerto inicial hasta el aeropuerto destino y todos los aeropuertos intermedios que visite así como la aeronave que se utilizara.
- 4.1.2.7. *Apertura de vuelos*: esta funcionalidad permitirá a los usuarios del aeropuerto confirmar que los pasajeros efectivamente viajaran y con cuantas maletas para posteriormente realizar la carga y balance del vuelo.
- 4.1.2.8. *Cierre de vuelos*: Una hora antes de la salida de cada vuelo se procederá a cerrar el vuelo esto con el fin de tener una lista oficial de los pasajeros finales que viajaran, cantidad de maletas.

4.1. Requerimientos NO funcionales del Sistema

- El Sistema debe ser una aplicación web (utilizando AngularJs, Bootstrap, HTML5 y CSS).
- La Base de Datos debe estar en MS Server.
- La capa de servicios debe estar desarrollada en C# y debe ser desplegada en el IIS.
- La Base de Datos debe estar al menos en tercera forma normal.
- Se deben implementar al menos 5 procedimientos almacenados y 2 triggers.
- El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

5. Entregables

- 5.1. Manual de Usuario.
- 5.2. Documentación Técnica y del proyecto (descrita en el punto 6).
- 5.3. Documento de instalación.
- 5.4. Plan de Proyecto.
- 5.5. Script de Base de Datos.
- 5.6. Script de población de Base de Datos.
- 5.7. Aplicación WEB.
- 5.8. Web Service/REST Service.
- 5.9. Minutas

6. Documentación

- 6.1. Se deberá documentar el código fuente.
- 6.2. Se deberá entregar un documento que contenga:
 - 6.2.1. Modelo conceptual utilizando la notación de Chen.
 - 6.2.2. Modelo relacional.
 - 6.2.3. Descripción de las estructuras de datos desarrolladas (Tablas).
 - 6.2.4. Descripción de los Store Procedures/Funciones/triggers implementados.
 - 6.2.5. Descripción detallada de la arquitectura desarrollada.
 - 6.2.6. Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
 - 6.2.7. Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
 - 6.2.8. Documentación de evidencia del trabajo en equipo.
 - 6.2.8.1. Actividades planeadas y su responsable. (Plan de trabajo)
 - 6.2.8.2. Minutas de sesiones de trabajo. (Seguimiento al plan de trabajo)
 - 6.2.8.3. Actividades realizadas por cada estudiante. (Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo,

llevan seguimiento de todo en el tiempo, imaginen que, si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.

6.2.8.4. Evidencia de uso de un manejador de código (se recomienda Gitgub).

6.2.9. Conclusiones y Recomendaciones del proyecto.

6.2.10. Bibliografía consultada en todo el proyecto

6.3. Diagrama de clases y un documento que explique el porqué del diseño.

7. Evaluación

1. El proyecto tendrá un valor de un 65% de la nota final, debe estar funcional.
2. Defensa del proyecto 15%.
3. La documentación tendrá un valor de un 20% de la nota final, cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
4. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código y Documentación.
5. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma OOP, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
6. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
7. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
8. De las notas mencionadas en los puntos 1 y 2 se calculará la Nota Final del Proyecto.
9. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
10. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - 10.1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - 10.2. Si no se utiliza un manejador de código se obtiene una nota de 0.
 - 10.3. Si no se entrega el punto 4 de la documentación se obtiene una nota de 0.
 - 10.4. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - 10.5. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - 10.6. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de 0.
 - 10.7. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.
11. Cada grupo tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
12. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
13. Cada grupo es responsable de llevar los equipos requeridos para la revisión.

14. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
15. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 14, no pueden participar en la revisión.
16. Después de enviada la nota final del proyecto el estudiante tendrá un máximo de 3 días hábiles para presentar un reclamo siempre y cuando la funcionalidad esté completa.

8. Referencias

AngularJS (2018-10-04). Recuperado de: <https://angularjs.io>

Bootstrap Themes & Templates (2018-10-04). Recuperado de: <https://wrapbootstrap.com/>

How to Write Doc Comments for the Javadoc Tool. (2018-10-04). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

C# Coding Conventions (C# Programming Guide). (2018-10-04). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>